**Amendments to the Specification:**

Please amend the title to read: Method and System for Dynamically Storing Frequently Used Instructions Using a Separate Cache

Please amend the paragraph on page 2, lines 17-20, as follows:

Therefore, it would be advantageous to have a method and apparatus that allows lines to continue to be cached based on the frequency of their use that can potentially increase the overall hit rates of cache memories.

Please amend the paragraph that starts on page 5, line 26, and ends on page 6, lines 26, as follows:

With reference now to **Figure 2**, a block diagram of a data processing system is shown in which the present invention may be implemented. Data processing system **200** is an example of a computer, such as computer **100** in **Figure 1**, in which code or instructions implementing the processes of the present invention may be located. Data processing system **200** employs a peripheral component interconnect (PCI) local bus architecture. Although the depicted example employs a PCI bus, other bus architectures such as Accelerated Graphics Port (AGP) and Industry Standard Architecture (ISA) may be used. Processor **202** and main memory **204** are connected to PCI local bus **206** through PCI bridge **208**. PCI bridge **208** also may include an integrated memory controller and cache memory for processor **202**. Additional connections to PCI local bus **206** may be made through direct component interconnection or through add-in boards. In the depicted example, local area network (LAN) adapter **210**[[,]] and small computer system interface SCSI host bus adapter **212**~~, and expansion bus interface 214~~ are connected to PCI local bus **206** by direct component connection. In contrast, audio adapter **216**, graphics adapter **218**, and audio/video adapter **219** are connected to PCI local bus **206** by add-in boards inserted into expansion slots. ~~Expansion bus interface 214 provides a connection for a keyboard and mouse adapter 220, modem 222, and additional memory 224.~~ SCSI host bus adapter **212** provides a connection for hard disk drive **226**, tape drive **228**, and CD-ROM drive **230**. Typical PCI local bus implementations will support three or four PCI expansion slots or add-in connectors.

Please amend the paragraph that starts on page 7, line 21, and ends on page 8, line 8, as follows:

For example, data processing system **200**, if optionally configured as a network computer, may not include SCSI host bus adapter **212**, hard disk drive **226**, tape drive **228**, and CD-ROM **230**~~, as noted by dotted line 232 in Figure 2 denoting optional inclusion~~. In that case, the computer, to be properly

called a client computer, must include some type of network communication interface~~, such as LAN adapter 210, modem 222, or the like~~. As another example, data processing system **200** may be a stand-alone system configured to be bootable without relying on some type of network communication interface, whether or not data processing system **200** comprises some type of network communication interface. As a further example, data processing system **200** may be a personal digital assistant (PDA), which is configured with ROM and/or flash ROM to provide non-volatile memory for storing operating system files and/or user-generated data.

Please amend the paragraph on page 8, lines 21-25, as follows:

The processes of the present invention are performed by processor **202** using computer implemented instructions, which may be located in a memory such as, for example, main memory **204**~~, memory 224,~~ or in one or more peripheral devices **226-230**.

Please amend the paragraph on page 9, lines 3-8, as follows:

In one embodiment, [[the]] each line in the main cache is outfitted with an associated counter that increments whenever that address line is accessed. When the counter reaches a certain number, the line is removed from cache and placed in the DFI-cache. The DFI-cache thereby holds more frequently accessed lines longer than main cache.

Please amend the paragraph that starts on page 9, line 16 and ends on page 10, line 4, as follows:

**Figure 3** shows a cache architecture of a computer system, consistent with a preferred embodiment of the present invention. In this illustrative example, two caches are depicted, first cache **302** (such as an instruction cache or I-cache) and Dynamic Frequent Instruction (DFI) cache **300**. First cache **302** of this example includes space for counters **308A-G** [[**308A-C**]] that correspond to each line **306A-G** [[**306A-C**]] of the I-cache **302**. Each line **306A-G** [[**306A-C**]] is outfitted with one such counter of counters **308A-G** [[**308A-C**]], and as a line, such as line **306A,** is accessed, its counter **308A** is increased. It should be noted that though this illustrative example makes reference to an I-cache as the first cache, other types of cache can be implemented in its place, such as victim cache as described by N. P. Jouppi in *"Improving Direct-Mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers,"* published in IEEE, CH2887-8/9/90, and hereby incorporated by reference.

Please amend the paragraph on page 15, lines 21-29, as follows:

In another embodiment, each line in the DFI-cache has a counter associated with it. When a line in the DFI-cache is hit, the values of counters associated with the other lines in the DFI-cache are

decremented. Thus, more frequently used lines in the DFI-cache have a higher value than less frequently used lines. When a line is to be replaced in the DFI-cache, the line with the lowest counter number is replaced. In this way, the DFI-cache holds ~~holes~~ frequently used lines longer than the I-cache.